# Directionally Unsplit Riemann-solver-based Hydrodynamic Schemes in Heterogeneous GPU Computing

**Hsi-Yu Schive (薛熙于)**
*Graduate Institute of Physics, National Taiwan University*
*Leung Center for Cosmology and Particle Astrophysics (LeCosPA)*

**Ui-Han Zhang (張瑋瀚), Tzihong Chiueh (闕志鴻),**
*Graduate Institute of Physics, National Taiwan University*
*Leung Center for Cosmology and Particle Astrophysics (LeCosPA)*

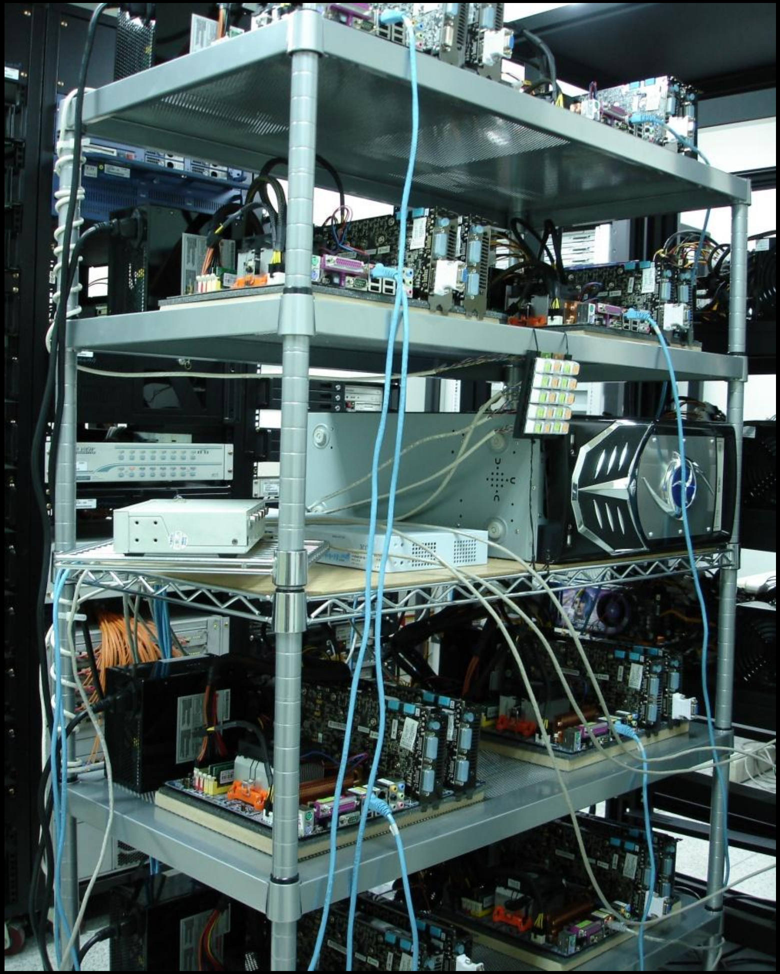
Manycore and Accelerator-based High-performance Scientific Computing Workshop
(27/01/2011 in ICCS)

# Outline

- **Introduction to GAMER**
  - GPU-accelerated Adaptive-MEsh-Refinement Code for Astrophysics
  - Previous benchmark results
- **Directionally unsplit hydro schemes**
  - MUSCL-Hancock Method (MHM)
  - Corner-Transport-Upwind (CTU)
- **Optimization and performance**
  - Uniform mesh
  - Adaptive mesh refinement
- **Conclusion and future work**

# Previous Works

- GraCCA system (2006)
  - Graphic-Card Cluster for Astrophysics
  - 16 nodes, 32 GPUs (GeForce 8800 GTX)
  - Peak performance: 16.2 TFLOPS
- Parallel direct N-body simulation in GraCCA
  - Individual time-step
  - 4th order Hermite integrator
  - 7.1 TFLOPS
  - GPU/CPU speed-up ~ 200



- Ref: Schive, H-Y., et al. 2008, NewA, 13, 418

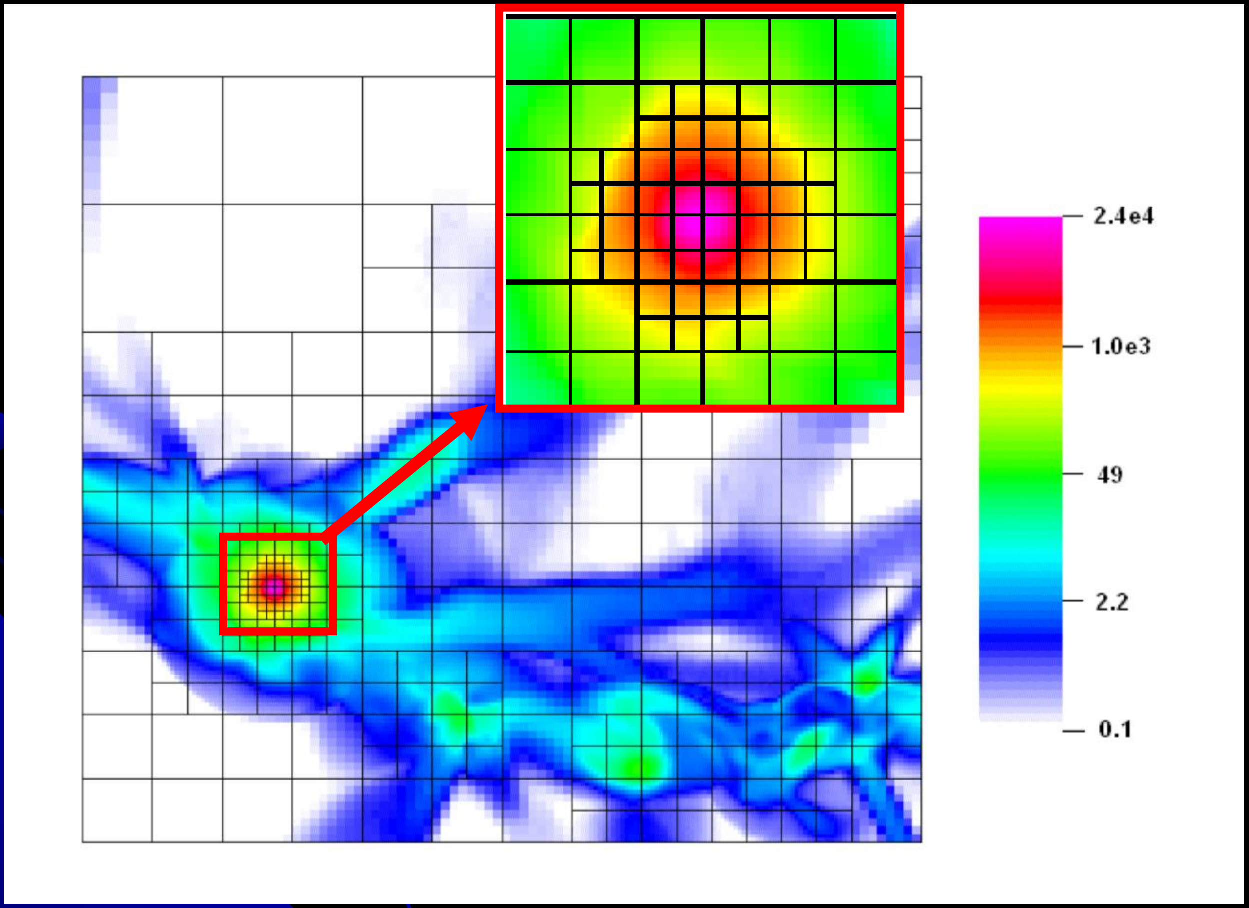
# GAMER

***GPU-accelerated Adaptive-MEsh-Refinement Code***

***for Astrophysics***

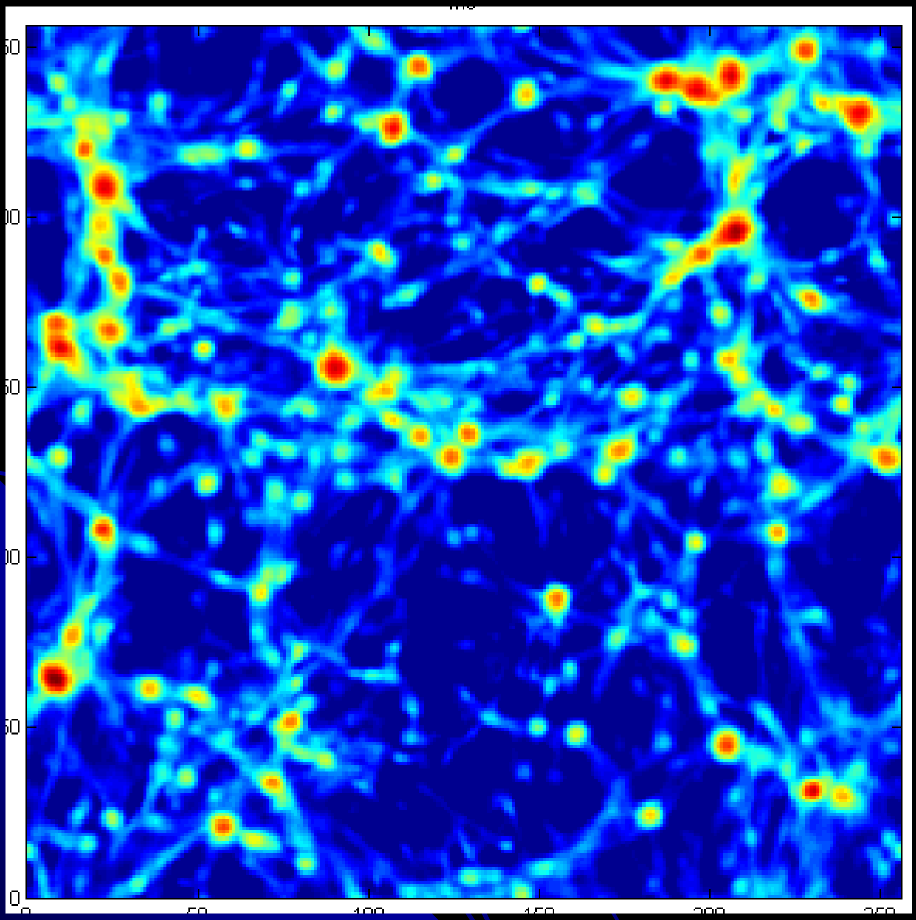
# Adaptive-Mesh-Refinement (AMR)

- Resolution adaptively changes with space and time
- Flexible refinement criteria



# AMR Example

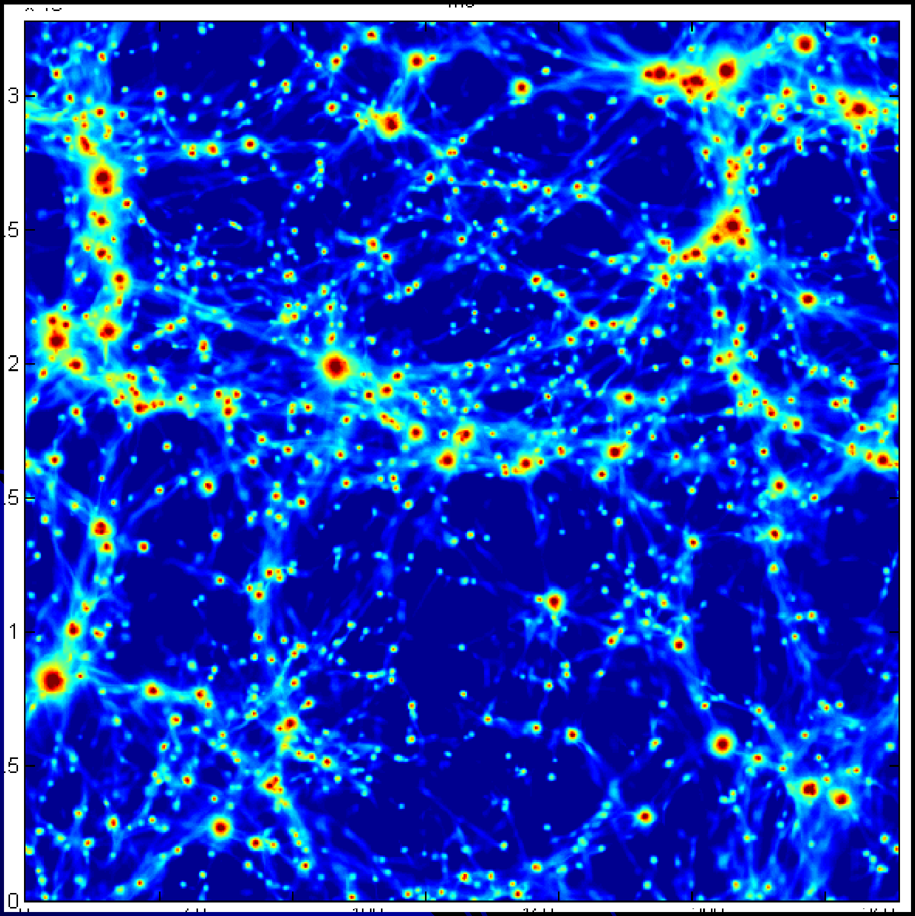
- Cosmology simulations with different refinement levels



Base level $256^3$, Refined level 0

# AMR Example

- Cosmology simulations with different refinement levels

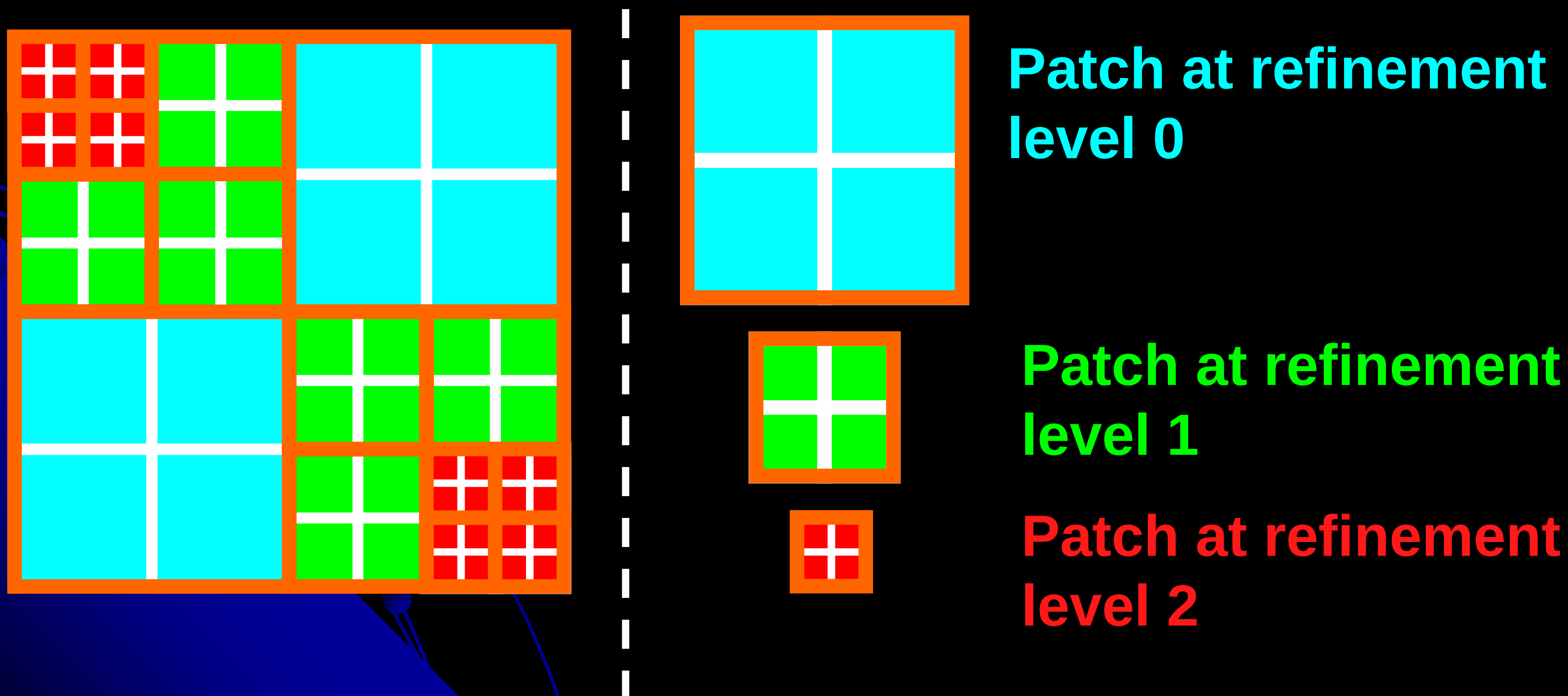


Base level $256^3$, Refined level 0

Base level $256^3$, Refined level 5 (effective resolution = $8192^3$)

# AMR Scheme in GAMER

- Refinement unit : patch (containing a fixed number of cells, e.g., $8^3$)
- Support GPU hydro and gravity solvers
- Hierarchical oct-tree data structure



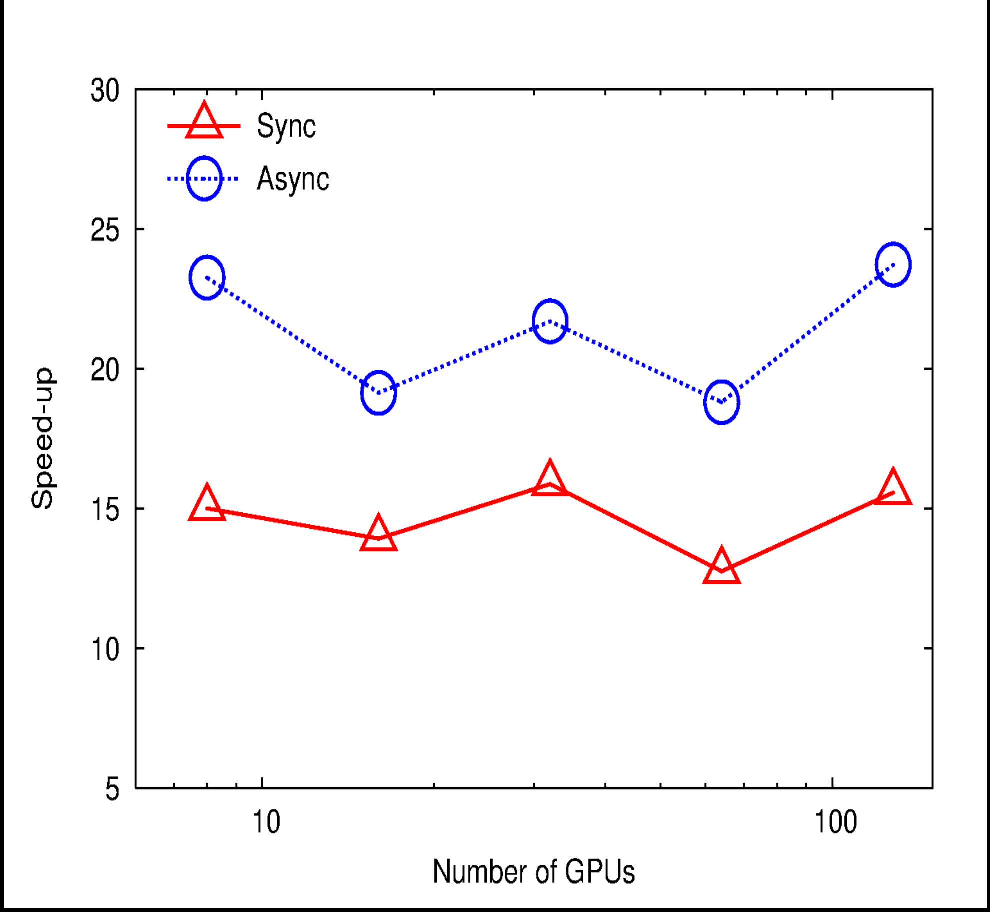
# Benchmark Test : NAOC GPU Cluster (Laohu)

National Astronomical Observatories, Chinese Academy of Sciences

1 – 128 Tesla C1060 vs.

1 – 128 Xeon E5520 cores

Speed-up is measured in <u>one-GPU-to-one-CPU-core</u> basis



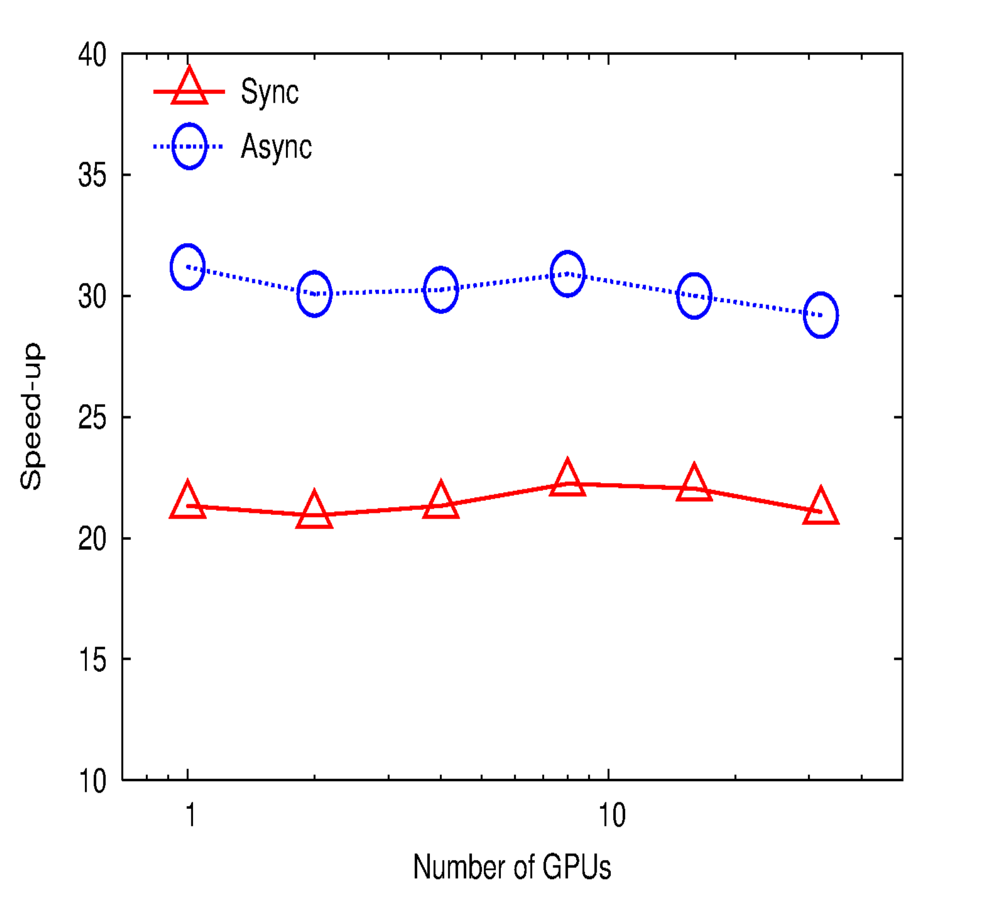
# Benchmark Test : NERSC GPU Cluster (Dirac)

National Energy Research Scientific Computing Center

1 – 32 Tesla C2050 vs.

1 – 32 Xeon E5530 cores

Speed-up is measured in one-GPU-to-one-CPU-core basis

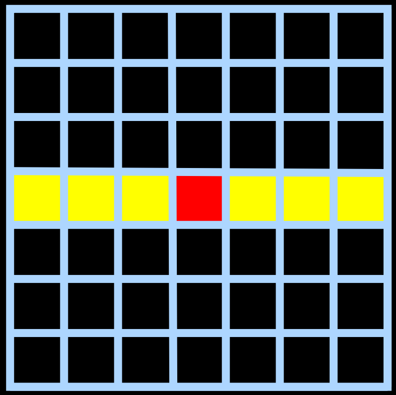


# Directionally Unsplit Hydro Schemes

# Splitting vs. Unsplitting

## Splitting methods :

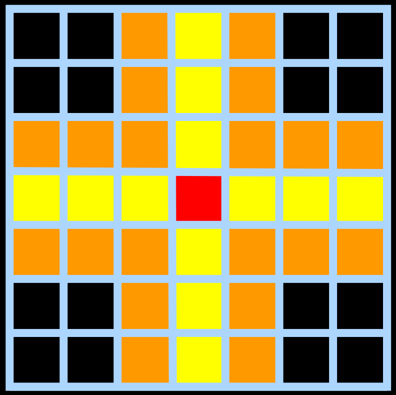
- 3D update : x→y→z
- 1D stencil



- GPU shared memory → straightforward
- Supported schemes in GAMER
  - Relaxing TVD (RTVD)
  - Weighted-Averaged-Flux (WAF)

## Unsplitting methods:

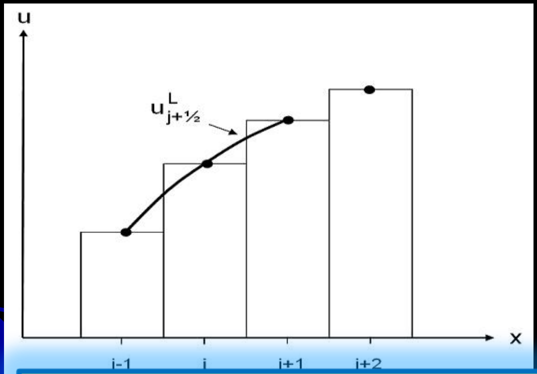
- 3D update : x + y + z
- 3D stencil



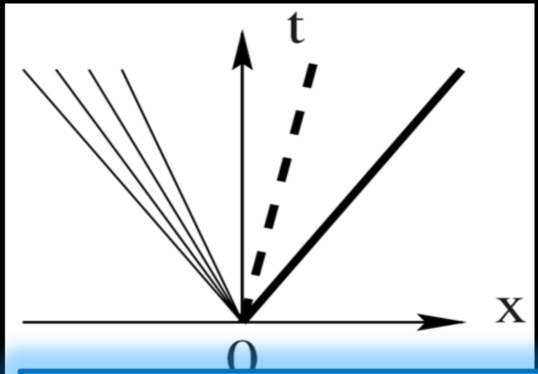
- GPU shared memory → non-trivial
- Supported schemes in GAMER
  - MUSCL-Hancock Method (MHM)
  - MUSCL-Hancock with Riemann prediction (MHMRP)
  - Corner-Transport-Upwind (CTU)

# Corner-Transport-Upwind (CTU)

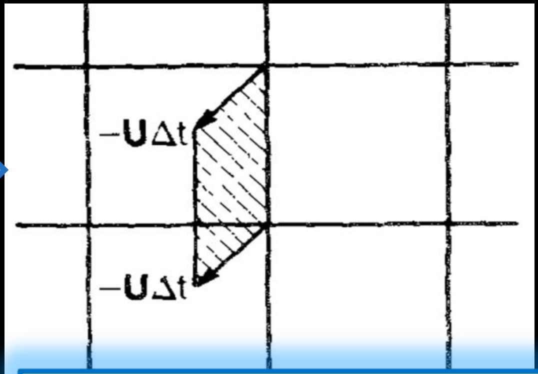
- **Colella, P., 1990. J. Comput. Phys. 87, 171.**
- **Extended to MHD and well tested in Athena code**
  - Stone, J.M., Gardiner, T.A., Teuben, P., Hawley, J.F., Simon, J.B., 2008. ApJS 178, 137.




**[1] Data reconstruction**
- Piecewise linear
- Piecewise parabolic
- Different slope limiters




**[2] Riemann solver**
- Exact solver
- Roe's solver
- HLLE/HLLC

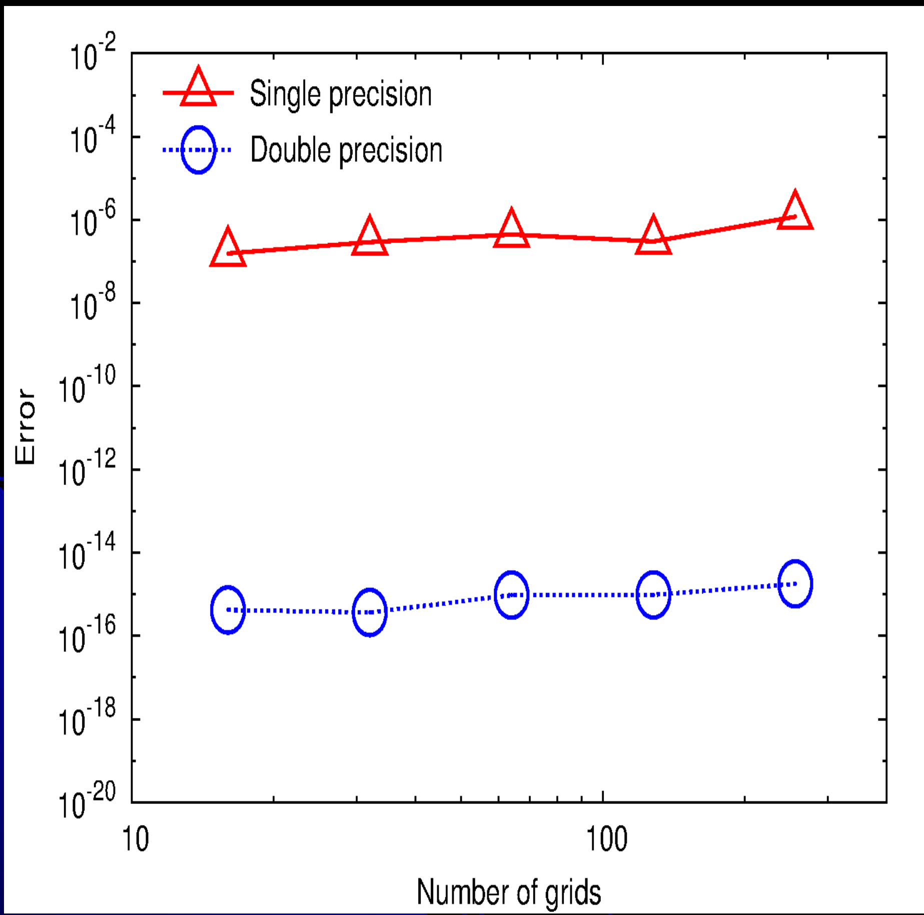



**[3] Transverse flux gradients**

**[4] Riemann solver**

**[5] Update solution**

# Accuracy Compared to Athena




**Athena :**

**Widely-adopted MHD/Hydro code (https://trac.princeton.edu/Athena/)**

**Developed by James M. Stone**

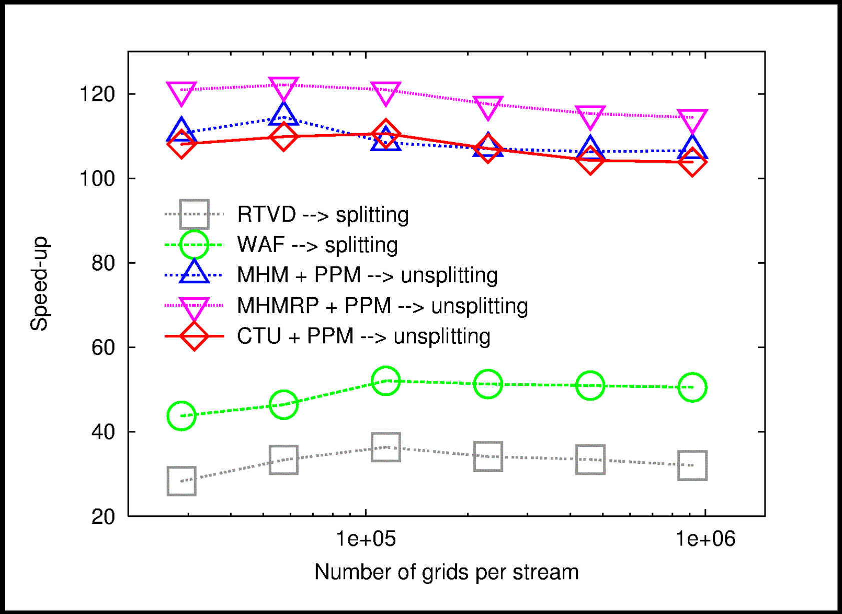
**Test problem : Blast-wave test**

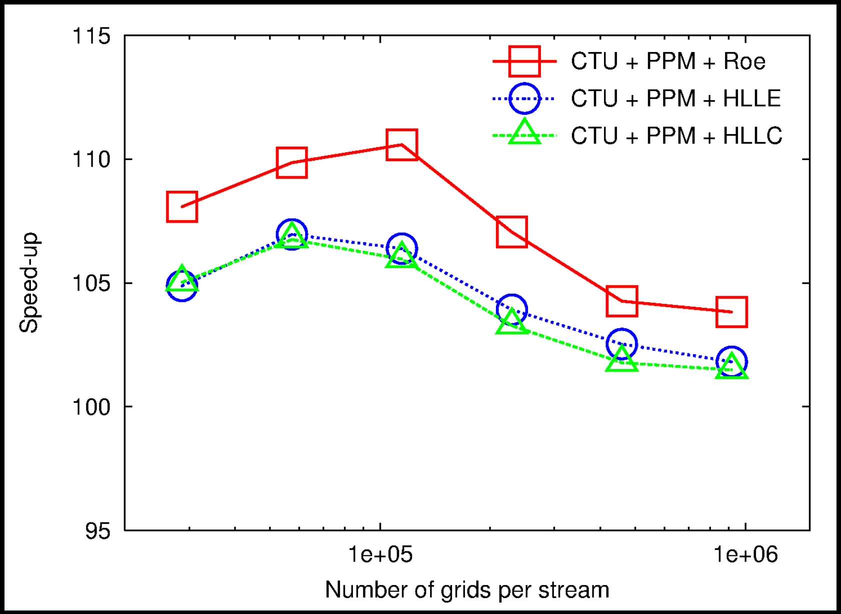
$$\text{Error} \equiv \frac{|\rho_{Athena} - \rho_{GAMER}|}{\rho_{Athena}}$$

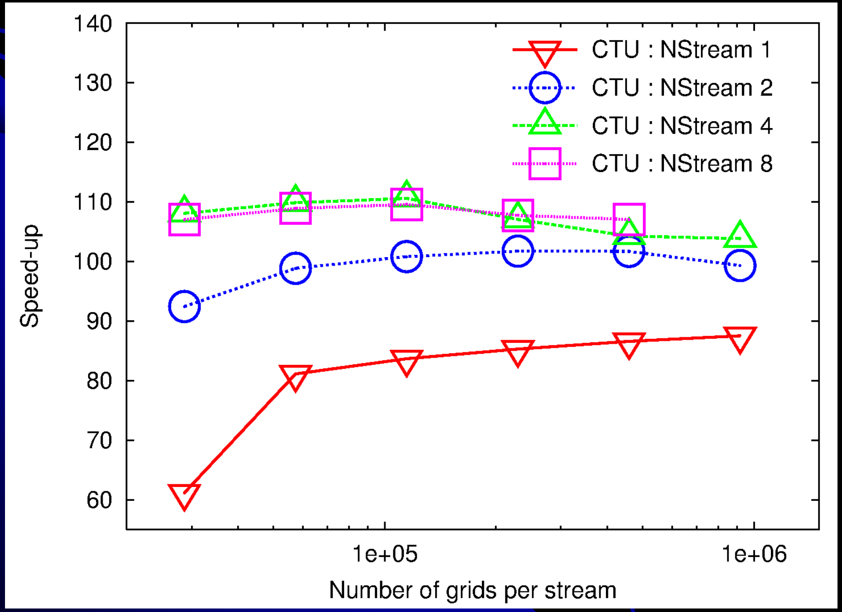
**$\text{Error}_{single}$ : $10^{-6}$ ~$10^{-7}$**

**$\text{Error}_{double}$ : $10^{-15}$~$10^{-16}$**

# Performance



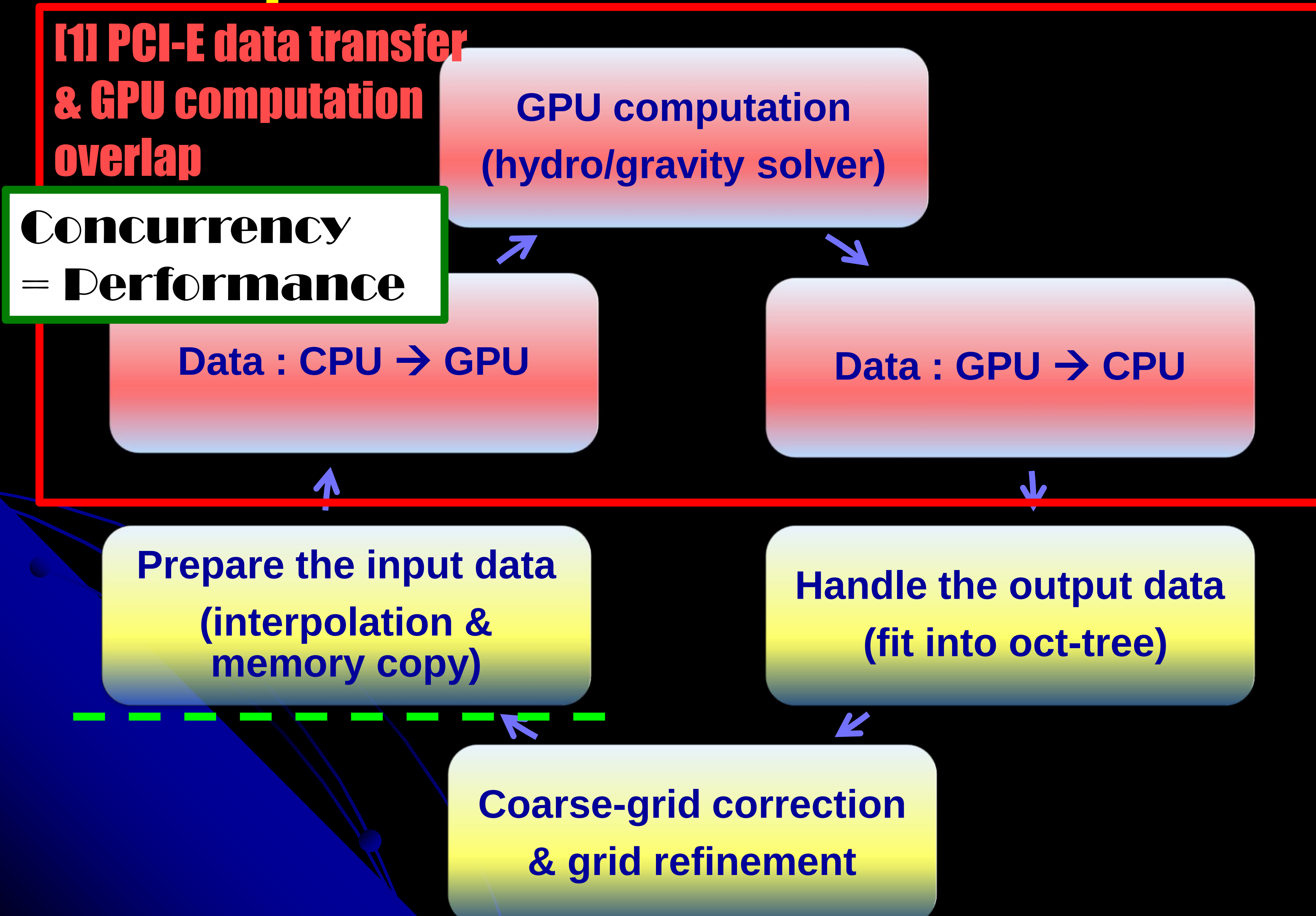

Splitting schemes → shared memory
Unsplitting schemes → global memory

CUDA stream : PCI-E / computation overlap

Speed-up : **110x ~ 120x**
(vs. 1 CPU core)

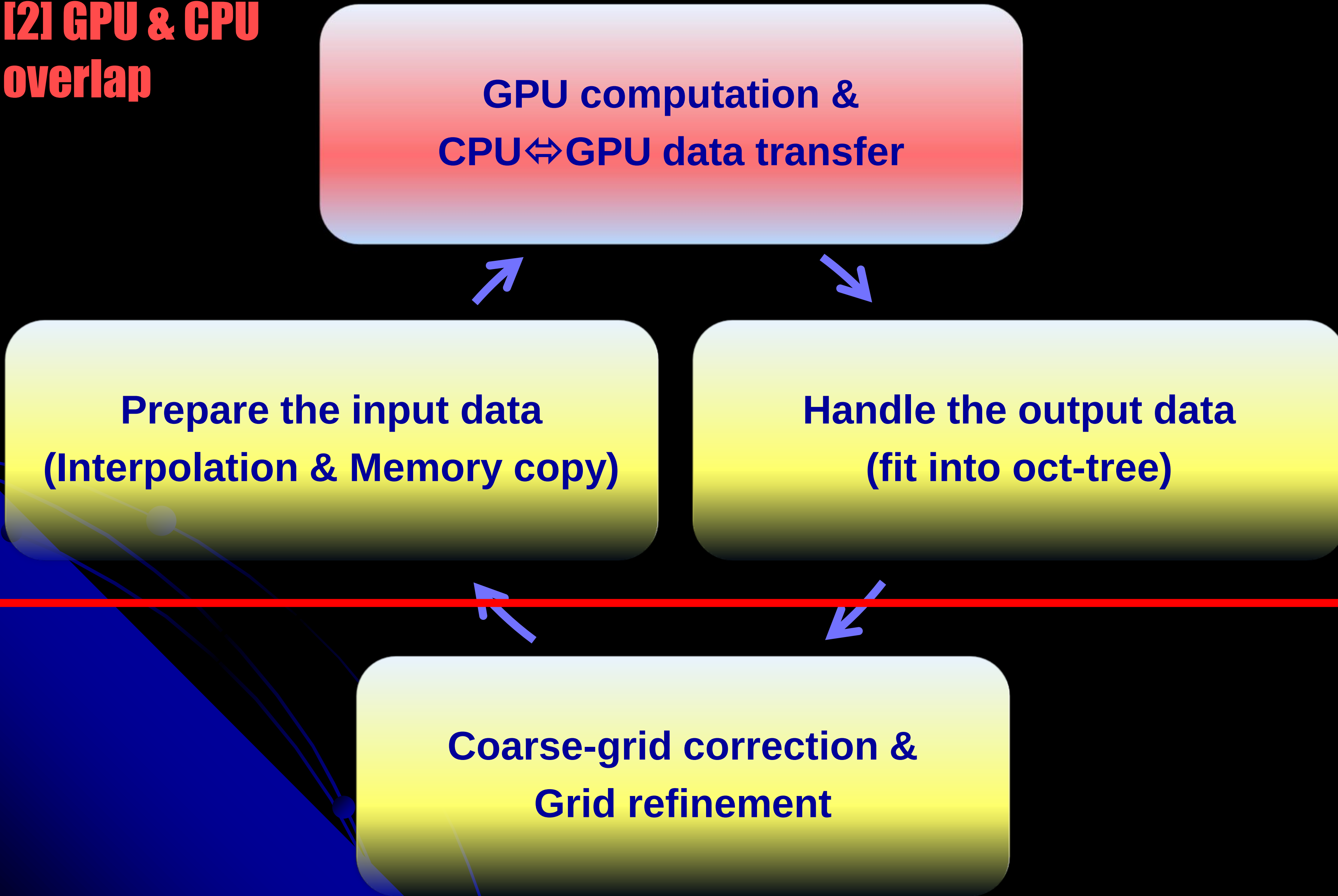
# Optimization and Performance in GAMER

# Optimization in GAMER



# Optimization in GAMER

**[2] GPU & CPU overlap**



# Optimization in GAMER

## [3] OpenMP parallelization

[CPU] Prepare the input data

OpenMP Thread 1
OpenMP Thread 2
…
OpenMP Thread N

+

[GPU] GPU computation & CPU⇔GPU data transfer

+

[CPU] Handle the output data

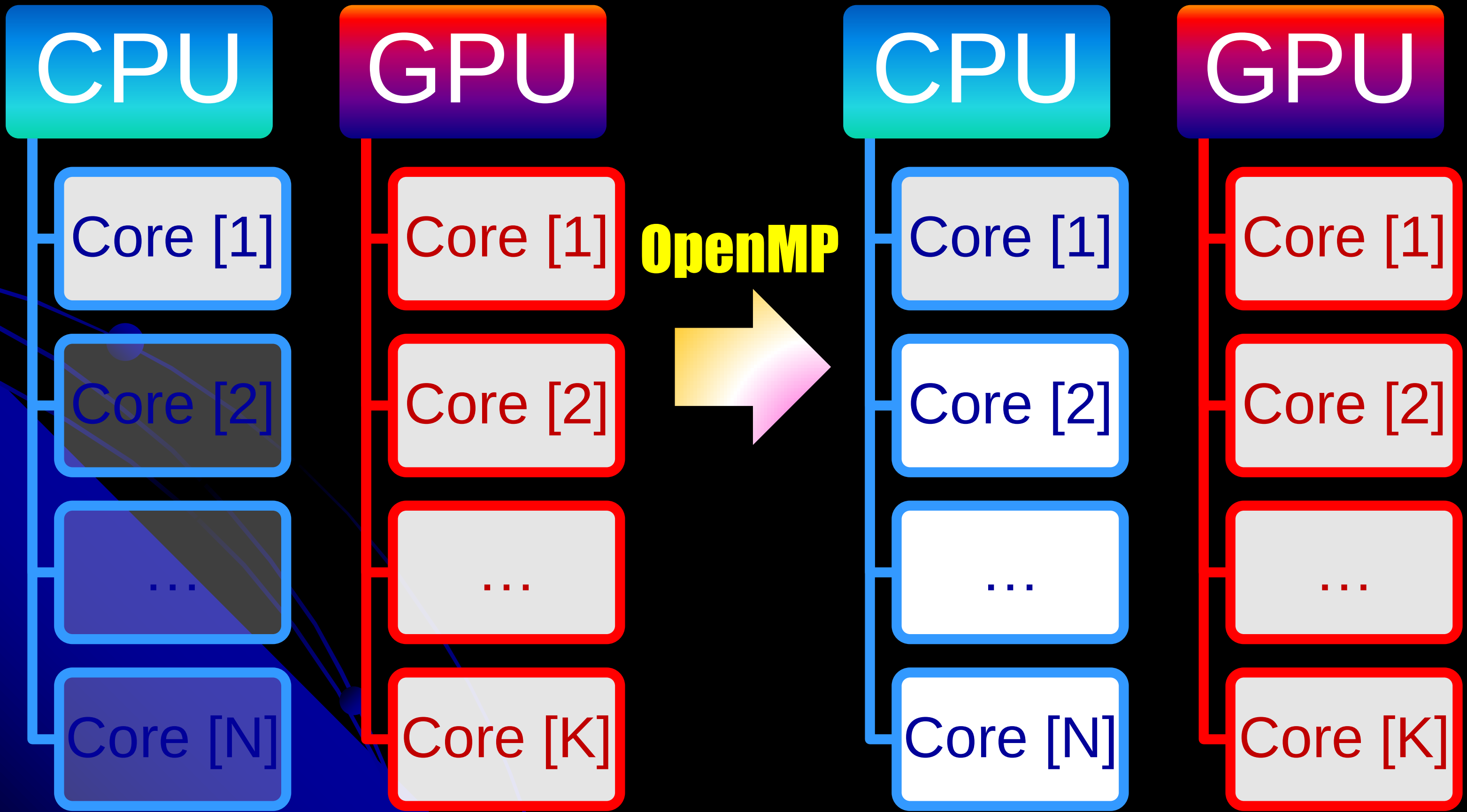
OpenMP Thread 1
OpenMP Thread 2
…
OpenMP Thread N

[CPU] Coarse-grid correction & Grid refinement

OpenMP Thread 1
OpenMP Thread 2
…
OpenMP Thread N

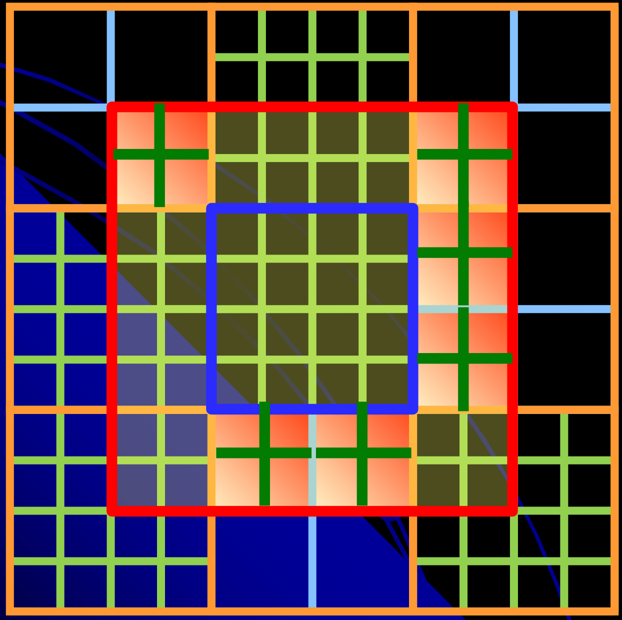
# OpenMP in GAMER

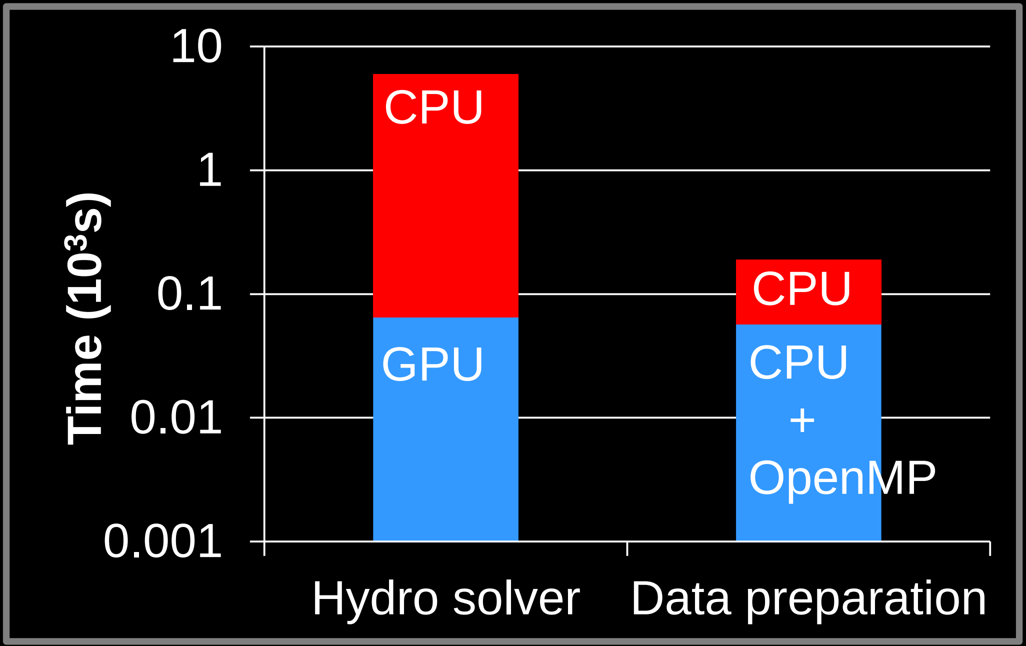
- Fully exploit the multi-core CPU computing power
  - N GPUs + K CPU cores (N≠K)



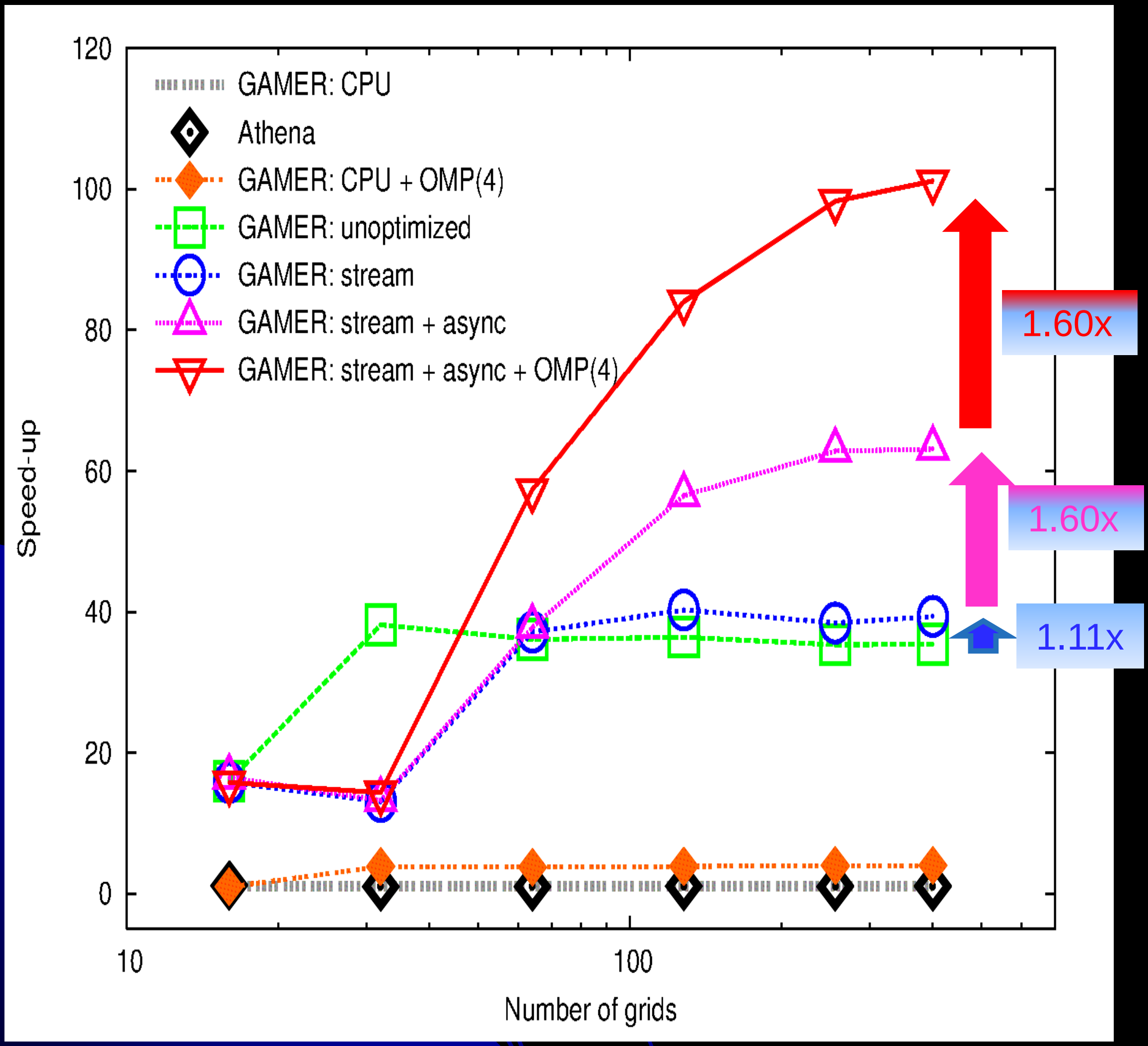
# OpenMP in GAMER

- **Performance can be significantly improved in GAMER**
  - **Using CPU to prepare the input data for GPU is extremely time-consuming**
  - **Performance bottleneck : 1.5x~3.0x longer than GPU calculation**





# Uniform-mesh Performance

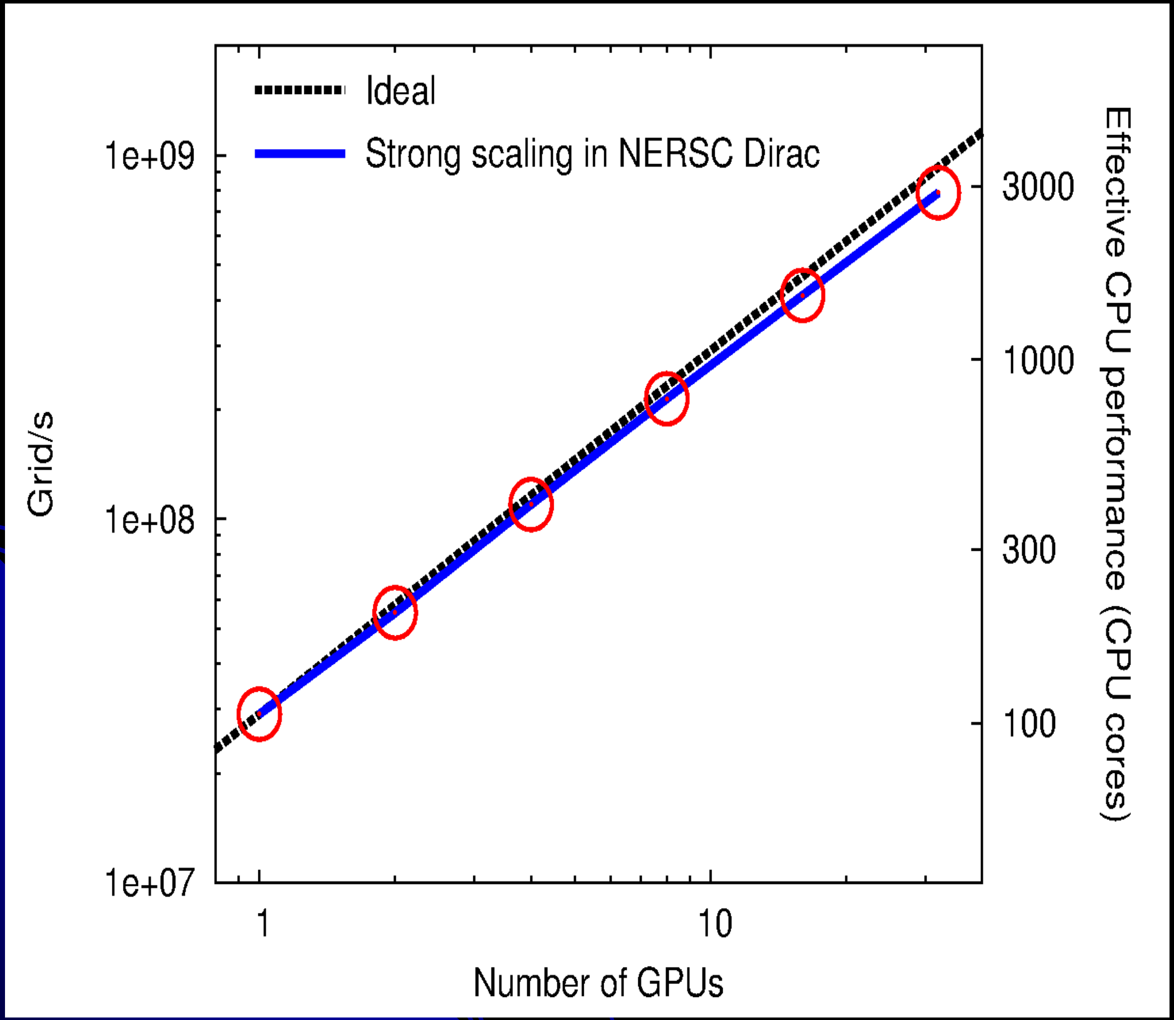



**GPU:** 1 NVIDIA Tesla C2050
**CPU:** 1 Intel Xeon E5530

**Stream : PCI-E/GPU overlap**
**Async : CPU/GPU overlap**
**OMP(4) : 4 OpenMP threads**

**GAMER-optimized vs.**
**1 CPU core : 101x**
**4 CPU cores: 25x**

# Strong Scaling



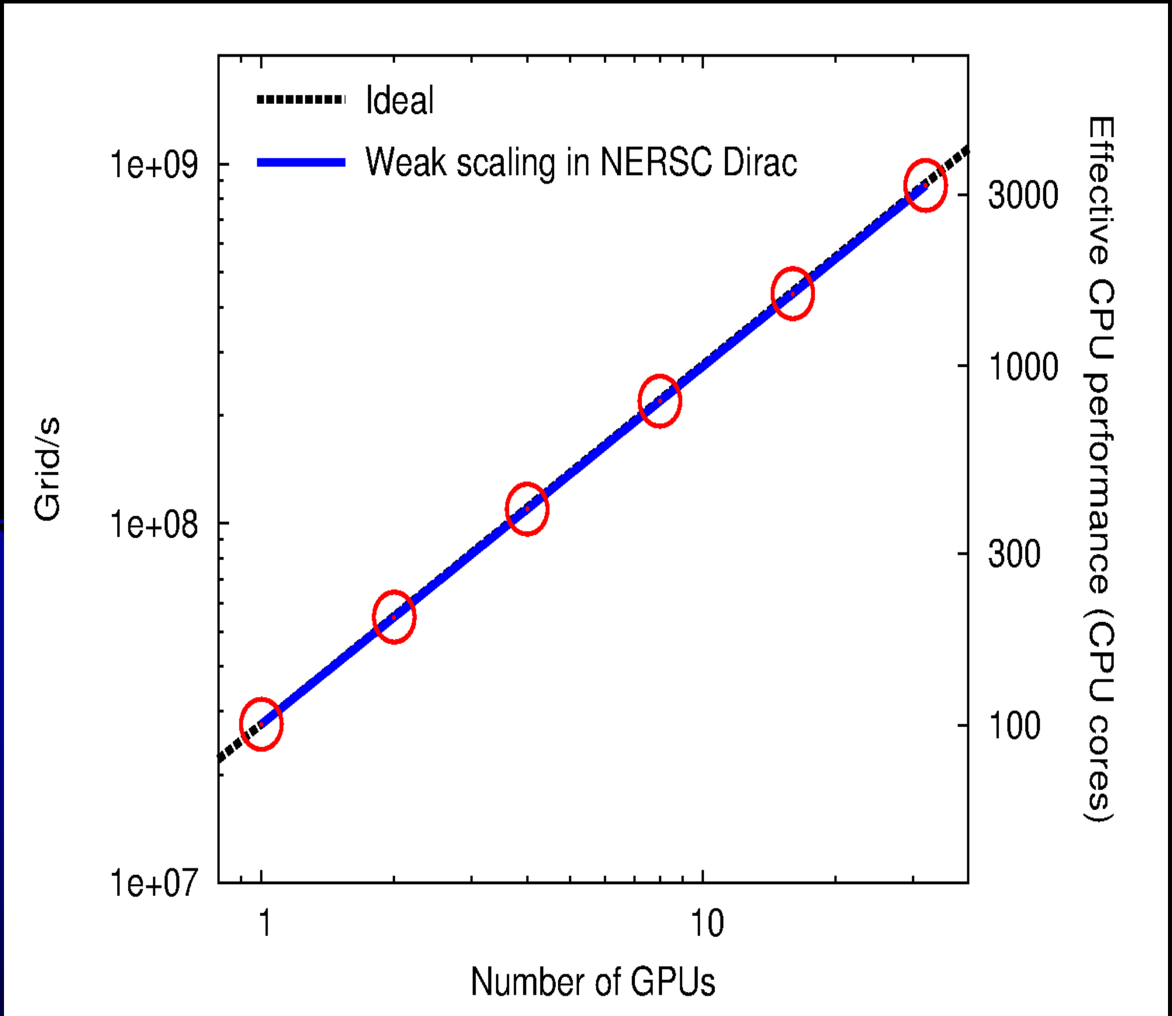

**Grid Size:**
**1024x1024x512**

**32-GPU Speed-up:**
**27.2x**

**Equivalent to:**
**2,751 CPU cores**

**Parallel Efficiency:**
**85.0%**

# Weak Scaling



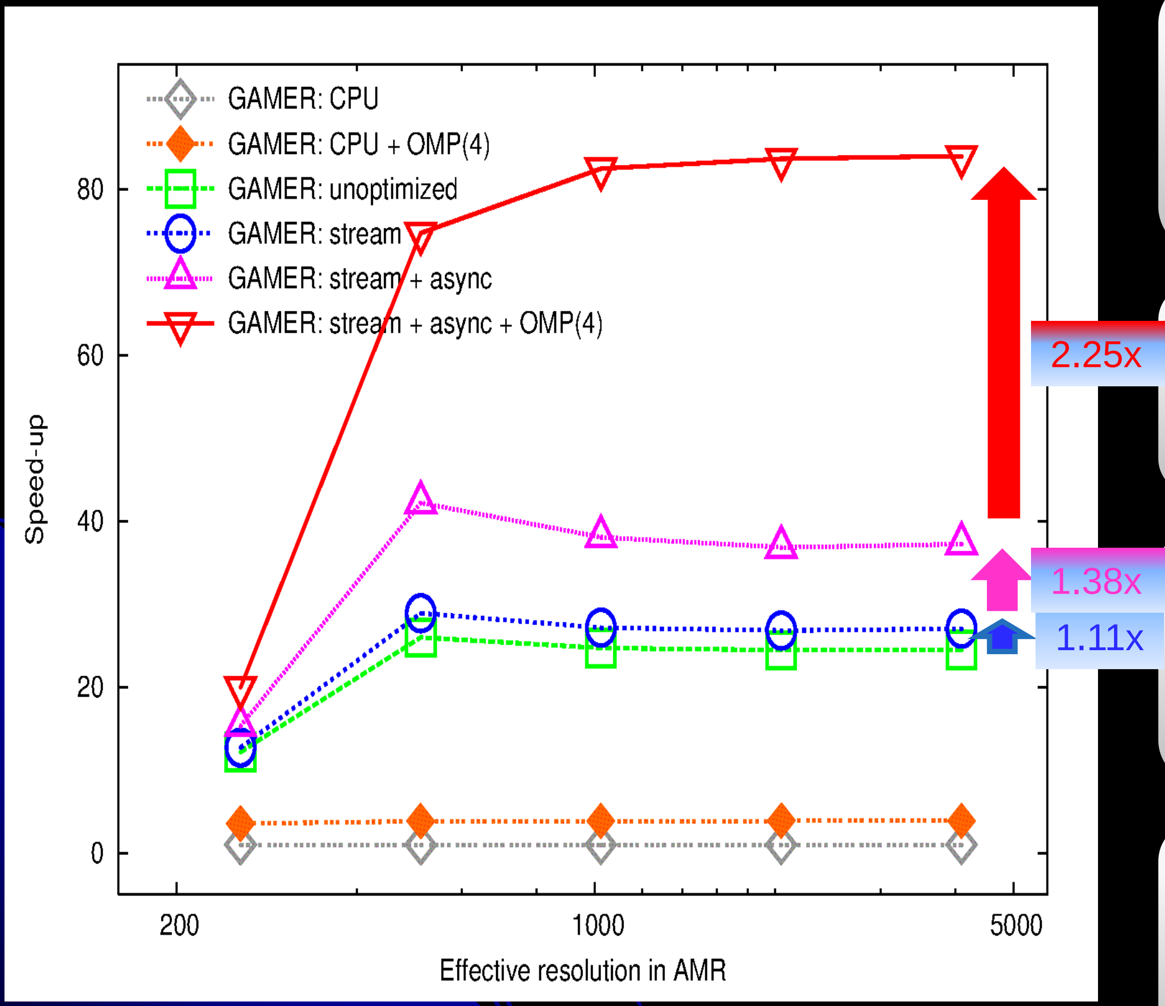

**Grid Size per GPU:** $512^3$

**32-GPU Speed-up:** 31.6x

**Equivalent to:** 3,189 CPU cores

**Parallel Efficiency:** 98.8%

# AMR Performance : Single GPU



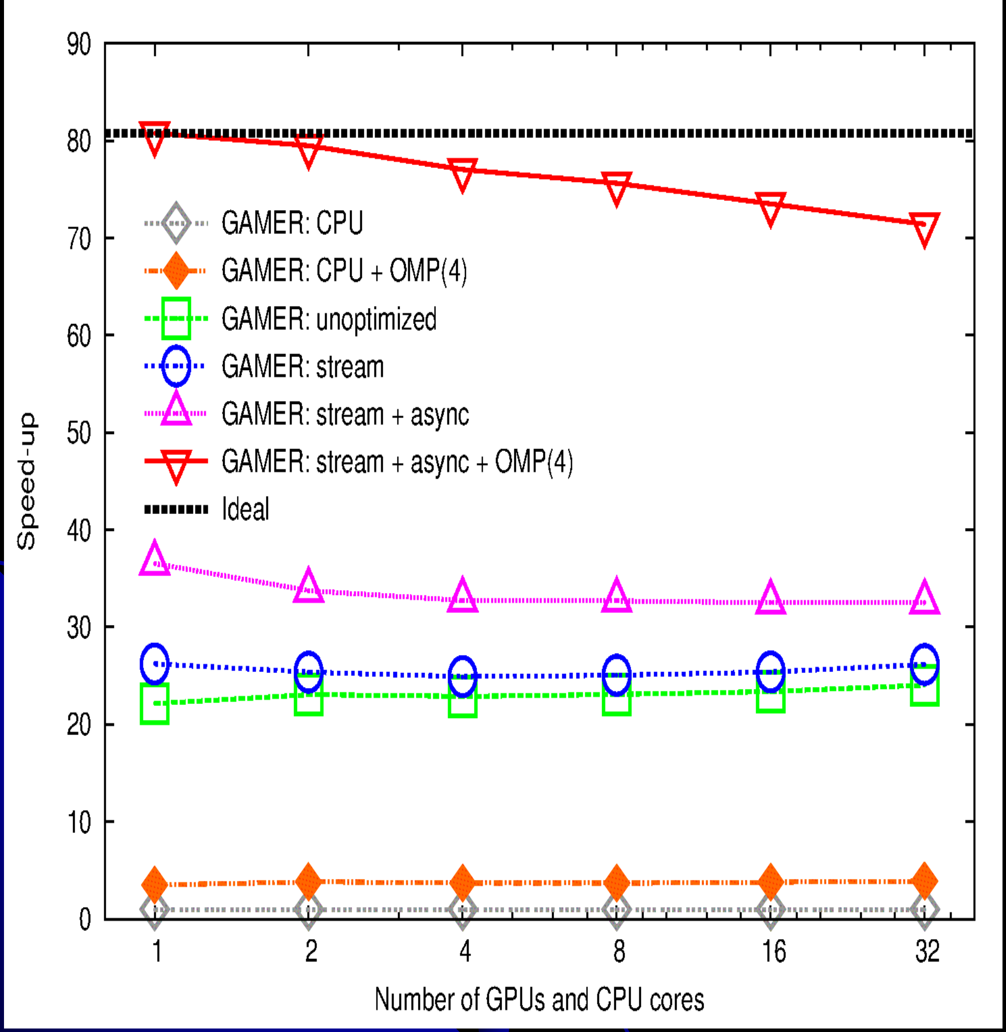

**NERSC Dirac GPU Cluster**
**GPU: 1 NVIDIA Tesla C2050**
**CPU: 1 Intel Xeon E5530**

**With self-gravity (80x speed-up in GPU) and individual time-step**

**Stream : PCI-E/GPU overlap**
**Async : CPU/GPU overlap**
**OMP(4) : 4 OpenMP threads**

**GAMER-optimized vs.**
**1 CPU core : 84x**
**4 CPU cores: 22x**

# AMR Performance : GPU Cluster




**NERSC Dirac GPU Cluster**

**GPU: 1-32 NVIDIA Tesla C2050**

**CPU: 1-32 Intel Xeon E5530**

**With self-gravity (80x speed-up in GPU) and individual time-step**

**Stream : PCI-E/GPU overlap**

**Async : CPU/GPU overlap**

**OMP(4) : 4 OpenMP threads**

**32 GPU vs. 32 CPU cores: 71x**

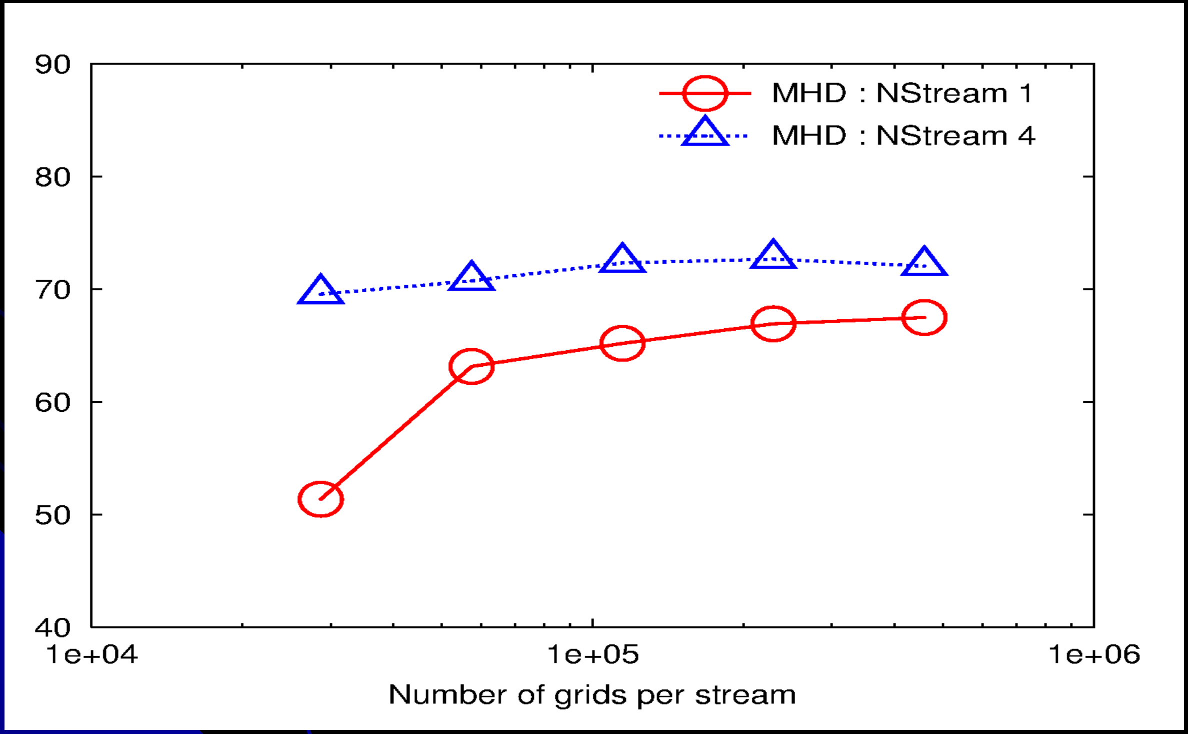
**32 GPU vs. 128 CPU cores: 18x**

**→ Equivalent to 2,304 CPU cores**

**MPI ~ 11% of $T_{total}$**

# Preliminary Results in Magnetohydrodynamics (MHD)

- Corner-transport-upwind scheme
  - Piecewise linear data reconstruction
  - Roe's Riemann solver extended to MHD
- Speed-up ~ 73x (still optimizing …)



# Conclusion

- **Directionally unsplit hydro schemes in GAMER :**
  - **Corner-Transport-Upwind & MUSCL-Hancock Method**
  - **MPI + OpenMP parallelization (multi CPUs + multi GPUs)**
  - **A framework of AMR + GPUs → general-purpose**
  - **80x ~ 100x speed-up (1 GPUs vs. 1 CPU core)**
  - **GAMER ref : Schive, H-Y., et al. 2010, ApJS, 186, 457**
- **Concurrency = Optimization**
  - **GPU computation**
  - **CPU ⇔ GPU data transfer**
  - **CPU computation**
  - **Multiple CPU cores (OpenMP)**
  - **Multiple GPUs (MPI)**

# Future Work

- More physics
  - MHD (coming soon)
  - Dark matter particles
- Overlap MPI time
- Load balance in AMR
  - Space-filling curve (Morton curve, Hilbert curve …)

  → Independent of the GPU implementation
- Complete analysis and visualization tools
- Desperate for more developers / users / applications
- Code request : b88202011@ntu.edu.tw